

# QoI-based Resource Allocation for Multi-Target Tracking in Energy Constrained Sensor Networks

Srikanth Hariharan  
Department of ECE  
The Ohio State University  
Columbus, OH, USA  
harihars@ece.osu.edu

Chatschik Bisdikian  
IBM T. J. Watson Research  
Hawthorne, NY, USA  
bisdik@us.ibm.com

Lance M. Kaplan  
Army Research Laboratory  
Adelphi, MD, USA  
lance.m.kaplan@us.army.mil

Tien Pham  
Army Research Laboratory  
Adelphi, MD, USA  
tien.pham1@us.army.mil

**Abstract**—Motivated by the need to judiciously allocate scarce sensing resources, and account for fusing information from multi-modal sensors, we develop a solutions methodology for maximizing the overall quality of information obtained subject to constraints on the energy utilized by a sensor network that is involved in the task of tracking multiple targets. Our methodology is based on integer programming, and explicitly allows for general fusion functions. We use an iterative Lagrangian relaxation technique to solve this problem where each iteration step involves solving for a Maximum Weight Independent Set (MWIS) of an appropriately constructed graph (which can be obtained in polynomial time for this problem). We apply our methodology to numerically study the problem of tracking targets moving over a period of time through a non-homogeneous, energy-constrained sensor field. In these applications, we study the QoI/energy trade-offs for various modes of operation including the period for measurement updates.

## I. INTRODUCTION

In a wireless sensor network, sensors with multiple modalities can be used to estimate a variety of features from objects of interest. For example, in target tracking, a radar can be employed to estimate the location and velocity of a target, while an imaging sensor can be employed to estimate its physical characteristics. Furthermore, fusing the information collected from the different modalities provides us with a more complete and accurate description providing a higher Quality of Information (QoI) [1], such as reducing the uncertainties regarding the tracks of the targets sensed.

Sensor networks can potentially perform multiple sensing and processing tasks at will. However, due to their limited bandwidth, energy, and computing resources, it becomes imperative to design and operate them in a way that is mindful of these limitations by judiciously allocating their resources to the tasks at hand.

In this paper, we are interested in a sensor network tracking multiple targets simultaneously. Specifically, we focus on ensuring that the QoI that is obtained by, say,

fusing information from multiple sensors can be represented as a general function of the QoIs obtained from individual sensors. This is of particular importance in heterogeneous, and multi-modal sensor networks. For example, the QoI obtained by fusing information from an imagery sensor (e.g., camera) and an acoustic sensor may not be a sum (or even a weighted sum) of the QoIs obtained by the individual sensors. Further, we take into account energy constraints from non-homogeneous sensors, where, for example, a camera may require more energy than an acoustic sensor.

There is extensive literature related to assignments in target-tracking scenarios based on integer programming formulations. In [2], Joshi et al., study how to select  $k$  out of  $m$  total sensors such that the error variance of the combined measurements of the  $k$  sensors is minimized while tracking a single target. Using convex relaxation techniques, they develop a solution for the case that both  $k$  and  $m$  are large. The proposed solution has limited application when multiple targets are present in the system. In [3] and [4], similar problems are studied for extended Kalman filters. The goal in [3] is to choose a group of sensors such that the total energy is minimized subject to constraints on the error variance, and that in [4] is to choose  $k$  out of  $m$  sensors in order to minimize the RMS position error. These studies have good scaling properties as  $k$  and  $m$  become large. However, in practical scenarios,  $k$  can be a small number. For instance, our numerical result in Figure 7(a) shows that we get diminishing returns in QoI as the number of sensors whose information is fused increases. Similar results have also been observed in [3]. Moreover, the sensors whose information are fused will typically be located closer to the target than other sensors. Hence, it is usually not necessary to solve the problem for a high value of  $k$  or  $m$ . Also, these studies are limited to specific estimation techniques (such as linear or extended Kalman filters). By relaxing the requirement that  $k$  and  $m$  are large, in this work, we develop a methodology that allows for general estimation and fusion techniques.

In [5], it is shown that the multi-target tracking problem is NP-Hard even when information can be fused from only two sensors; it also provides approximation algorithms for the problem by observing a relationship to a bin-packing problem. This study does not consider sensor resource limitations. In [6], this problem is studied under constraints on the

Research was sponsored by US Army Research laboratory and was accomplished under Agreement Number W911NF-06-3-0002-P00013, and ARO MURI Award W911NF-07-10376. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, or the U.S. Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>JUL 2011</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2011 to 00-00-2011</b>	
4. TITLE AND SUBTITLE <b>QoI-based Resource Allocation for Multi-Target Tracking in Energy Constrained Sensor Networks</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>U.S. Army Research Laboratory, Adelphi, MD, 20783-1197</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>Presented at the 14th International Conference on Information Fusion held in Chicago, IL on 5-8 July 2011. Sponsored in part by Office of Naval Research and U.S. Army Research Laboratory.</b>					
14. ABSTRACT <b>Motivated by the need to judiciously allocate scarce sensing resources, and account for fusing information from multi-modal sensors, we develop a solutions methodology for maximizing the overall quality of information obtained subject to constraints on the energy utilized by a sensor network that is involved in the task of tracking multiple targets. Our methodology is based on integer programming, and explicitly allows for general fusion functions. We use an iterative Lagrangian relaxation technique to solve this problem where each iteration step involves solving for a Maximum Weight Independent Set (MWIS) of an appropriately constructed graph (which can be obtained in polynomial time for this problem). We apply our methodology to numerically study the problem of tracking targets moving over a period of time through a non-homogeneous, energy-constrained sensor field. In these applications, we study the QoI/energy tradeoffs for various modes of operation including the period for measurement updates.</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>8</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

reward required for each target, and a solution is obtained by modeling the problem as a semi-matching problem. In both these studies, a sensor can track only one target, but a target can be tracked by multiple sensors. Also, they assume that the QoI (for fusion) obtained from multiple sensors can be expressed as the sum of the QoIs obtained from individual sensors. Therefore, while this work takes some constraints on the quality into account, it does not capture general fusion metrics for QoI as considered in this paper.

We show that without energy constraints, our problem maps to a Maximum Weight Independent Set (MWIS) problem. Though, finding an MWIS in a general graph is NP-Hard, we show that we can find such a set in polynomial time for this problem because of the structure of the graph in which it is obtained. When energy constraints are present, a particularly challenging problem in integer programming, we use Lagrange multipliers to formulate the dual problem and provide an iterative solution, which again involves an MWIS.

MWIS techniques have been used for associating sensor measurements to targets in [7], [8]. This is a fundamentally different class of assignment problems than what we consider in our framework. Moreover, these studies do not consider operational constraints such as energy. Hence, we are dealing with graphs whose structure is different from those in these studies.

Lagrangian techniques have been extensively studied for continuous optimization problems. However, due to duality gaps, they have limited applications in integer optimization. Therefore, it is important to understand where these techniques can be applied in this regard. In a generalized assignment problem (GAP) [9] with linear budget constraints, using a Lagrange multiplier for the budget constraint results in a dual problem which can be solved using iterative techniques such as the primal-dual sub-gradient algorithm. This work provides a comprehensive tutorial of this approach. While this approach does not always guarantee an optimal solution for the primal problem, a number of studies in integer programming theory derive an optimal primal solution leveraging Lagrangian techniques [10], [11]. We discuss these issues carefully in Sections IV and V.

Considering the operation of a system over slotted time, our main contributions in this paper are:

- the development of a general integer programming methodology for allocating a constrained pool of sensors to one or more targets in each slot, that allows for a system-level QoI function for fusing information from multiple sensors as a general function of the QoIs from individual sensors for each target, and energy constraints;
- an iterative solution for the integer programming problem using a primal-dual gradient projection algorithm, which involves solving an MWIS problem at each iteration in an appropriately constructed graph;
- an analysis of the convergence properties, and duality gaps of the algorithm; and
- an extensive performance, simulation, and trade-off evaluation while tuning system parameters for improving the Quality of Information (QoI) for the multi-target

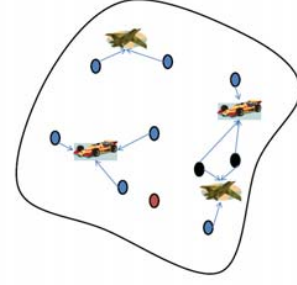


Figure 1. Model

tracking problem.

The rest of the paper is organized as follows. In Section II, we describe the system model. In Section III, we put forth our methodology for assigning sensors to multiple targets. In Section IV, we develop an iterative algorithm to solve the problems formulated in Section III, and discuss analytical results on optimality and convergence. In Section V, we quantify our analytical results through extensive numerical evaluations. We finish in Section VI with some concluding remarks.

## II. SYSTEM MODEL

We consider a sensor network comprising a set of sensors  $S$  of size  $N$  and a set of targets  $T$  of size  $M$ . We assume that the network has detected and associated measurements to targets using appropriate techniques such as Joint Probabilistic Data Association (JPDA), clustering algorithms, etc. We consider time-slotted operation with sensors tracking targets during each *sampling instant* (or *sampling period*). Based on predicted QoI in the next sampling slot, a group of sensors is *assigned* to track each target to maximize the overall QoI obtained in that sampling slot while satisfying energy constraints. The QoI, for instance, can be the predicted Fisher information obtained from a Kalman filter. An example of our system model is provided in Figure 1. The black sensors track multiple targets during the same sampling instant, the blue sensors track only one target, and the red sensor tracks no targets during that sampling instant.

## III. PROBLEM MODEL

In this section, we formulate a model (referred to as *Target-Oriented Model (TOM)*) for the problem at hand. In this model, we focus on sensor assignments for the *next* sampling slot. By updating QoIs and solving this problem for each future slot, we can track multiple targets over a time horizon. Analogous to this model, we can also have a *Sensor-Oriented Model*; details can be found in [12].

### A. Target-Oriented Model (TOM)

According to this model, a target is tracked by a group of sensors. Specifically, for each target  $i \in T$ ,  $i = 1, \dots, M$ , we associate a collection of sets  $\mathbb{K}_i$  that represents the possible groups of sensors that can be assigned simultaneously to that target. Let  $\mathbb{K}_i = \{K_1^i, K_2^i, \dots, K_{m_i}^i\}$  where for each

$l \in \{1, \dots, m_i\}$ ,  $K_l^i \subseteq \{0\} \cup S$ . The “0” element represents the case that target  $i$  is not assigned to any sensor (during a slot). Clearly, the “0” element can be a member of only one of the sets in  $\mathbb{K}_i$  and that set must be a singleton. The introduction of the  $\{0\}$  set in  $\mathbb{K}_i$  allows us to explicitly model a (potential) penalty for not assigning target  $i$  to any sensor. A penalty is relevant when the QoI is (say) the error variance of a target, and the goal is to minimize the sum of the error variances. We use this in later sections.

For each  $i \in T$ , and  $j \in \mathbb{K}_i$ , we define  $x_{ij}$  to be the assignment indicator variable:

$$x_{ij} = \begin{cases} 1, & \text{if target } i \text{ is assigned to sensor group } j; \text{ and} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Let  $q_{ij}$  represent the QoI obtained from target  $i$  when  $x_{ij} = 1$  holds true, i.e., when target  $i$  is tracked by *all* the sensors in the set  $j \in \mathbb{K}_i$  (and only these sensors). Likewise, let  $e_{ij}$  denote the energy demand for tracking target  $i$  when  $x_{ij} = 1$  holds true, and let  $e$  be the total energy constraint on the system. Now the optimization problem  $\Pi_T$  for TOM can be formulated as follows:

**Problem  $\Pi_T$ :** For  $x_{ij} \in \{0, 1\}$ , maximize

$$\sum_{i \in T} \sum_{j \in \mathbb{K}_i} q_{ij} x_{ij}, \text{ s.t. (a) } \sum_{j \in \mathbb{K}_i} x_{ij} = 1 \text{ for each } i \in T; \\ \text{and (b) } \sum_{i \in T} \sum_{j \in \mathbb{K}_i} e_{ij} x_{ij} \leq e. \quad (2)$$

The objective of  $\Pi_T$  is to maximize the sum of the QoIs obtained by all the targets in the system. Constraint (a) is a matching constraint and it states that each target  $i$  must be assigned to exactly one of the sets of sensors in  $\mathbb{K}_i$ , which by construction contains all permissible alternatives for assigning target  $i$  to the sensors (even the  $\{0\}$  set). Constraint (b) is the energy constraint.

Note that that  $q_{ij}$ , i.e., the QoI obtained when sensors in the set  $j \in \mathbb{K}_i$  track target  $i$ , can be an *arbitrary* function of the QoIs obtained from each sensor separately. This implies that TOM can be used even when, say, the QoI attained by fusing information from multiple sensors is not restricted to be the sum of the individual QoIs as was the case in [5], [6]. Further, note that the models developed in [2], [3], [6] are all special cases of the model developed in this section.

#### B. Extensions to Problem $\Pi_T$

##### 1. A sensor can track only one target but a target can be tracked by multiple sensors

We can model this constrained problem by including the following linear constraint in problem  $\Pi_T$ :

$$\text{for each sensor } k \in S, \sum_{i \in T} \sum_{j \in \mathbb{K}_i: k \in j} x_{ij} \leq 1 \quad (3)$$

This constraint implies that a sensor  $k$  can be assigned to track at most one target  $i \in T$ . This is because for each set  $j$  containing the sensor  $k$ , and for each target  $i$ , at most one of the  $x_{ij}$ s can be one. Our numerical results study this extension.

##### 2. Minimize the total energy subject to QoI constraint $q$

This problem is the dual problem of  $\Pi_T$ . The solution to this problem can be found by combining a binary search to the solution of  $\Pi_T$  [12].

##### 3. Guarantee that certain targets achieve at least a given minimum QoI

Suppose that target  $i$  requires a QoI of at least  $q$ . The solution to this problem can be obtained by adding the following constraint to  $\Pi_T$ : for each  $j \in \mathbb{K}_i$ , if  $q_{ij} < q$ , set  $x_{ij} = 0$ .

##### 4. Guarantee that certain sensors satisfy a given maximum energy

Suppose that sensor  $k$  can expend at most  $e_k$  units of energy. This can be modeled by the following linear constraint:

$$\sum_{i \in T} \sum_{j \in \mathbb{K}_i: k \in j} x_{ij} e_{ij}^{(k)} \leq e_k, \quad (4)$$

where  $e_{ij}^{(k)}$  is an arbitrary function of  $e_{ij}$  representing the energy contribution of sensor  $k$  in the set  $j$  of sensors. Note that  $e_{ij}$  represents the energy required by *all* the sensors in the set  $j$  to track the target  $i$ , hence  $e_{ij} = \sum_{l \in j} e_{ij}^{(l)}$ . This problem can be solved in the same manner (using a Lagrange multiplier for this constraint) as  $\Pi_T$  is solved in the next section.

#### IV. SOLUTIONS METHODOLOGY

In this section, we develop iterative algorithms for problem  $\Pi_T$ , and discuss optimality and convergence results. We first find the optimal solution for the problem without energy constraints. Then, we use this to find an optimal solution for the problem with energy constraints.

##### A. Without Energy Constraints

We show that an optimal solution can be obtained by finding an MWIS. An *independent set* in a graph is a set of nodes no two of which have an edge between them. A Maximum Weight Independent Set (MWIS) is an independent set with maximum total weight of nodes. Finding an MWIS in a general graph is NP-Hard—it follows from the fact that finding a maximum clique in a general graph is NP-Hard [13]. Nonetheless, in our problem, due to the structure of the graph involved, it is not NP-Hard to find such a set.

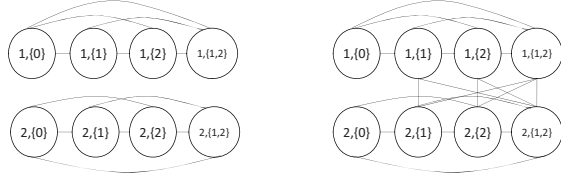
**Theorem 4.1:** Optimal solutions to problem  $\Pi_T$  (without energy constraints) can be obtained by finding an MWIS in the graph  $\mathcal{G}$  constructed as follows:

- for each variable  $x_{ij}$ , create a node  $(i, j)$  in  $\mathcal{G}$ ;
- for each node  $(i, j)$  in  $\mathcal{G}$ , associate a weight  $q_{ij}$ ; and
- for every two nodes  $(i_1, j_1)$  and  $(i_2, j_2)$ , create an edge between them if and only if  $i_1 = i_2$ .

**Proof:** From the above construction, two nodes  $(i_1, j_1)$  and  $(i_2, j_2)$  in  $\mathcal{G}$  can both be in an independent set of  $\mathcal{G}$  if and only if  $i_1 \neq i_2$ . This implies that the variables  $x_{i_1 j_1}$  and  $x_{i_2 j_2}$  can both equal 1 simultaneously if and only if  $i_1 \neq i_2$ . Therefore, for TOM, an independent set cannot consist of any two nodes in  $\mathcal{G}$  that correspond to the same target.

Since by finding an MWIS, we maximize  $\sum_{i \in T} \sum_{j \in \mathbb{K}_i} q_{ij} x_{ij}$  while satisfying the constraints, this





(a) The graph  $\mathcal{G}$  is a disjoint union of cliques (b) Graph  $\mathcal{H}$ : A sensor can track only one target

Figure 2. Graph construction for  $\Pi_T$

MWIS provides an optimal solution to problem  $\Pi_T$  (without energy constraints). ■

**Corollary 4.1:** When a sensor can track only one target (in a given sampling slot), an optimal solution to TOM (without energy constraints) can be obtained by finding an MWIS in the graph  $\mathcal{H}$ , where  $\mathcal{H}$  is constructed as follows:

- construct  $\mathcal{G}$  as in Theorem 4.1; and
- for any two nodes  $(i_1, j_1)$  and  $(i_2, j_2)$  where  $\{0\} \notin j_1$  and  $\{0\} \notin j_2$ , create an edge between them if and only if  $j_1 \cap j_2 \neq \emptyset$ .

*Proof:* In the case where a sensor can track only one target, we have all the original constraints in problem  $\Pi_T$  (apart from the energy constraint), and the additional constraint given in (3).  $\mathcal{G}$  represents all the constraints except the constraint in (3). Since for any node  $(i_m, j_m)$ ,  $j_m$  represents the set of sensors tracking target  $i_m$ , if  $j_1 \cap j_2 \neq \emptyset$ , one of the sensors in  $j_1$  and  $j_2$  is tracking both targets  $i_1$  and  $i_2$ . Therefore, we create an edge between these two nodes. Further, we do not create an edge in  $\mathcal{G}$  if  $j_1$  and  $j_2$  do not have any common sensor. Therefore, an independent set in  $\mathcal{H}$  will satisfy the required constraints, and hence an MWIS in  $\mathcal{H}$  will provide an optimal solution for this case in the absence of energy constraints. ■

We now show that an MWIS in  $\mathcal{G}$  can be found in polynomial time.

**Theorem 4.2:**  $\mathcal{G}$  is a union of disjoint cliques, and hence an MWIS in  $\mathcal{G}$  can be found in polynomial time.

*Proof:* By the construction of  $\mathcal{G}$  in Theorem 4.1, two nodes  $(i_1, j_1)$  and  $(i_2, j_2)$  will have an edge between them if and only if  $i_1 = i_2$ . Hence, there is no edge between these nodes when  $i_1 \neq i_2$ . Therefore, the set of nodes having the same first label  $i_1$  form a clique, and, thus,  $\mathcal{G}$  is a union of cliques. Since there are no edges between node from different cliques,  $\mathcal{G}$  is a union of disjoint cliques.

Hence, an MWIS in  $\mathcal{G}$  is obtained by simply selecting the node in each clique with maximum weight. This can clearly be obtained in polynomial time. ■

Figure 2(a) illustrates the result in Theorem 4.2 for the case of two sensors and two targets. The node label  $(1, \{1, 2\})$  means that target 1 is tracked by both sensors 1 and 2. It can be easily seen that  $\mathcal{G}$  is a disjoint union of cliques.

In Figure 2(b), we provide an example of the special case where a sensor can track only one target. The graph  $\mathcal{H}$  is not a disjoint union of cliques. For instance, we can see that there

is an edge between  $(1, \{1, 2\})$  and  $(2, \{1\})$  since if target 1 is tracked by both sensors 1 and 2, then target 2 cannot be tracked by sensor 1. When the number of targets is a constant, the complexity of finding an MWIS in this graph is polynomial in the number of nodes in the graph. The reason is as follows: Since we can only select one group of sensors for each target, if there are  $t$  targets in the system, and  $n$  nodes in this graph, the complexity of finding an MWIS in this graph is  $O((n/t)^t)$ . Note that  $n/t = |\mathbb{K}|$ .

### B. With Energy Constraints

We now take the energy constraints into account as well. As explained in the extensions in Section III, it is straightforward to account for other energy constraints such as per-sensor energy constraints. Associating a Lagrange multiplier  $\lambda$  for the energy constraint, the dual objective function for  $\Pi_T$  can be obtained as follows, where  $x_{ij} \in \{0, 1\}$ :

$$D(\lambda) = \max_{(2.a)} \left\{ \sum_{i \in T} \sum_{j \in \mathbb{K}_i} (q_{ij} - \lambda e_{ij}) x_{ij} \right\} + \lambda e, \quad (5)$$

where the (2.a) qualifier for the “max” operator refers to constraint (a) in the maximization formulation for  $\Pi_T$  in (2). From (5), it can be immediately seen that for a given  $\lambda$ ,  $D(\lambda)$  can be obtained by finding an MWIS in  $\mathcal{G}$  where the weight of each node  $(i, j)$  in  $\mathcal{G}$  (see Theorem 4.1) is modified from  $q_{ij}$  to  $(q_{ij} - \lambda e_{ij})$ .

The dual optimization problem of  $\Pi_T$  is given by

$$\min_{\lambda \geq 0} D(\lambda). \quad (6)$$

We solve this problem using a gradient projection algorithm.

**Algorithm 1:** Initialize  $\lambda = 0$ , and:

- 1) At iteration  $k$ , compute  $\vec{x}^{(k)} = \{x_{ij}^{(k)} : i \in T, j \in \mathbb{K}_i\}$  as

$$\vec{x}^{(k)} = \arg \max_{(2.a)} \left\{ \sum_{i \in T} \sum_{j \in \mathbb{K}_i} (q_{ij} - \lambda^{(k-1)} e_{ij}) x_{ij} \right\},$$

where  $x_{ij} \in \{0, 1\}$ . This can be computed by finding an MWIS in  $\mathcal{G}$  with the weights modified as described before.

- 2) At iteration  $k$ , update  $\lambda$  as follows.

$$\lambda^{(k)} = [\lambda^{(k-1)} + \alpha^{(k)} (\sum_{i \in T} \sum_{j \in \mathbb{K}_i} e_{ij} x_{ij}^{(k)} - e)]^+, \quad (7)$$

where  $[y]^+ = \max\{0, y\}$ . The coefficient  $\alpha^{(k)}$  is a positive step-size used at iteration  $k$  and it can be chosen according to Theorem 4.3 later on. One of the possible choices of  $\alpha^{(k)}$  is  $1/k$ .

- 3) Stop when  $\lambda^{(k)} - \lambda^{(k-1)} < \gamma$ , where  $\gamma$  is a threshold.

It follows from step 2 that  $\lambda$  increases when the energy required by the system is greater than  $e$ , and it decreases when the energy required by the system is less than  $e$ . Hence, a binary search algorithm can be used instead for updating  $\lambda$  as well. This algorithm results in much faster convergence than the gradient projection algorithm. However, the gradient projection algorithm is very useful when multiple Lagrange multipliers need to be updated simultaneously (corresponding to multiple constraints).

### C. Computational Complexity

We consider the binary search algorithm for updating  $\lambda$ . The number of iterations required by the binary search algorithm is  $O(\lceil \log_2(d/\gamma) \rceil)$ , where  $\gamma$  is the threshold in Algorithm 1, and  $d = e_{max}$  is the maximum energy that the system can utilize, i.e., the system uses all the sensors available.

(i) *A sensor can track multiple targets*: The complexity of finding an MWIS in this case is simply the number of nodes in the graph  $\mathcal{G}$ . Assume that there are  $s$  sensors,  $t$  targets, and that at most  $k$  of the sensors can be combined for a target, e.g., information from at most  $k$  out of  $s$  sensors can be fused for sensing a target. Then the number of nodes in  $\mathcal{G}$  for TOM is at most  $t \times (1 + s + \binom{s}{2} + \dots + \binom{s}{k}) \sim O(ts^k)$ . Further, since this computation needs to be performed at most  $\lceil \log_2(d/\gamma) \rceil$  times (for binary search), the overall complexity is given by  $O(ts^k \lceil \log_2(d/\gamma) \rceil)$ .

(ii) *A sensor can track only one target*: In this case, we use the result from the end of Section IV-A and the fact that for case (i),  $|\mathbb{K}| \sim s^k$ , to obtain the complexity of finding a MWIS as  $O(s^{kt})$ . Therefore, the overall complexity with energy constraints is  $O(s^{kt} \lceil \log_2(d/\gamma) \rceil)$ .

**Remark:** As we mentioned earlier, this work is motivated by the fact that  $s$ ,  $t$  and  $k$  are small numbers in practice. While our algorithms have a polynomial time complexity, they may not scale well as  $s$ ,  $t$ , or  $k$  become large. This is a trade-off that we obtain for allowing general fusion functions.

### D. Optimality and Convergence

If the  $x_{ij}$ s were relaxed to take continuous values in  $[0, 1]$ , then for diminishing step-sizes ( $\alpha^{(k)} \rightarrow 0$ ,  $\sum_{k=1}^{\infty} \alpha^{(k)} \rightarrow \infty$ ), as  $k \rightarrow \infty$ , Algorithm 1 would converge to an optimal solution for both the primal and dual problems, and there would be no duality gap [14]. However, since the  $x_{ij}$ s are indicator ( $\{0,1\}$ -valued) variables, we would need post-processing to obtain valid values for the  $x_{ij}$ s. It is, in general, very hard to analytically compare this post-processed integer solution with the optimal integer programming solution.

Here, we do not relax the integer variables  $x_{ij}$ . In this case, while the dual program is a linear optimization program for  $\lambda$ , the primal program is an integer optimization program. Therefore, there could potentially exist a duality gap. The following result shows the convergence of  $\lambda$ .

**Theorem 4.3:** For any  $\epsilon > 0$ , for diminishing step-sizes ( $\alpha^{(k)} \rightarrow 0$ ,  $\sum_{k=1}^{\infty} \alpha^{(k)} \rightarrow \infty$ ),  $\exists B > 0$  such that  $\forall k > B$ ,  $|\lambda^{(k)} - \lambda^*| < \epsilon$ , where  $\lambda^*$  is the optimal solution to the dual problem.

The proof follows from [15].

While  $\lambda$  converges to the optimal solution of the dual problem, we cannot guarantee that the primal objective function converges. Suppose that the primal objective function does not converge. When the algorithm stops, we will obtain two values of  $\lambda$ ,  $\lambda_1$  and  $\lambda_2$ , where, say,  $\lambda_1 < \lambda_2$ , and  $\lambda_2 - \lambda_1 < \gamma$ .  $\lambda_1$  will result in an infeasible solution to the primal problem, while  $\lambda_2$  will result in a feasible solution to the primal problem. This has been proved in [10], [11] and we also observed it during our numerical evaluations in Section V. One way to find the primal

optimal solution is to perform a branch and bound starting with these solutions that we obtained from Algorithm 1. Recently, it was shown in [10] that given a  $\rho$ -approximation algorithm for Step 1 of Algorithm 1, a  $(\frac{\rho}{\rho+1} - \epsilon)$ -approximation algorithm for the primal problem can be obtained using the feasible and infeasible solutions obtained either by binary search or by Algorithm 1. In our case, since the MWIS problem can be solved in polynomial time because of the graph structure, using Algorithm 1 and the algorithm provided in [10], we can obtain a  $(0.50 - \epsilon)$ -approximation for our primal problem.

## V. NUMERICAL RESULTS

We now investigate a multi-target tracking problem with Kalman filters and study the performance of the QoI obtained; we use the variance of the track estimate as the QoI. As we mentioned before, different types of filters, fusion techniques, QoI metrics can be employed. We use Kalman filters here for the purpose of illustrating how our methodology can be applied. Even with these basic filters, we obtain insightful results on how QoI behaves with energy.

### A. Setup

We consider a system with  $M = 3$  targets and  $N = 9$  sensors, three of which are high energy ( $h$ ) sensors and six are low energy ( $l$ ) ones. A sensor can track only one target during a sampling instant, but a target can be tracked by multiple sensors. Information is required from two low energy sensors, or one high energy sensor, or a combination of one high energy sensor and two low energy sensors in order to estimate the location of a target. The mobility models for the targets are given by the following equations (we use a scalar formulation for ease of presentation):

$$x_i(k+1) = a_i x_i(k) + v_i(k), \quad i \in \{1, 2, 3\}, \quad (8)$$

where  $x_i(k)$  is the location of target  $i$  at slot  $k$  and  $v_i(k)$  is AWGN with distribution  $\mathcal{N}(0, Q_i)$ . The measurement model for each of the high energy and low energy sensors is (we drop the sensor index for brevity):

$$z_i^e(k) = x_i(k) + y_i^e(k), \quad e \in \{h, l\}, \text{ and } i \in \{1, 2, 3\}, \quad (9)$$

where  $y_i^e(k)$  is AWGN with distribution  $\mathcal{N}(0, R_i^e)$ . We assume that  $R_i^e$ ,  $e \in \{h, l\}$ , depends on the distance between the sensor and the target. Specifically, the measurement error variance is 4 for a high energy sensor and 9 for a low energy sensor, if the distance between the sensor and the target is less than 10 units. The measurement error variance increases by 1 if the distance is between 10 and 20 units, and increases by 2 if the distance is more than 20 units.

The locations of the sensors and the initial locations of the targets are random. Table I provides the numerical values of the parameters in the setup. The parameters  $e_h$ ,  $e_l$ ,  $e_{ll}$ , and  $e_{hll}$  represent the energy levels used by a high energy sensor only, a low energy sensor only, a combination of two low energy sensors, and a combination of one high energy and two low energy sensors, respectively.

Parameter	Value	Parameter	Value
$a_1$	1.5	$Q_1$	4
$a_2$	1.2	$Q_2$	6
$a_3$	1.1	$Q_3$	4
$R^h$	4	$e_h$	2
$R^l$	9	$e_l$	1
$e_{hll}$	6	$e_{ll}$	2.4

Table I  
SIMULATION PARAMETERS

Let  $P_i^p(k)$  represent the one-step predicted variance for target  $i$  at slot  $k$ , and  $P_i^c(k)$  represent the corrected variance after the measurement for target  $i$  at slot  $k$  is received. The variance update of the Kalman filters are given by the following equations,  $k = 1, 2, \dots$  ( $i = 1, 2, 3$ ):

$$P_i^p(k) = a_i^2 P_i^c(k-1) + Q_i, \text{ where } P_i^c(0) = 0, \quad (10)$$

$$P_i^c(k) = \frac{P_i^p(k) \cdot R_i^e}{P_i^p(k) + R_i^e} \Leftrightarrow \frac{1}{P_i^c(k)} = \frac{1}{P_i^p(k)} + \frac{1}{R_i^e}, e \in \{h, l\}. \quad (11)$$

Thus, when “knowledge” is added, which in this case is represented by a newly arriving measurement with variance  $R_i^e$ , the tracking variances before and after incorporating the new knowledge exhibit a harmonic relationship; see also [16]. We adopt a similar relationship for the QoI obtained when fusing across track estimates from different sensors as well. Alternative fusion expressions could also be used without altering the applicability of our methodology.

With this setup, and with our interest in minimizing the sum the QoI (variance) over all the targets, we now study a number of performance metrics such as the convergence of our algorithms, the performance of the overall variance with energy, and over a time horizon.

### B. Algorithm Convergence

We start with an experimental study of the convergence for the algorithms developed in Section IV. Note that while the Lagrange multiplier always converges, the primal objective function may not converge. Nonetheless, it would eventually remain constrained to within  $\epsilon$  from the optimal solution for some  $\epsilon > 0$ . We study two cases, one in which the primal objective function converges, and one where it does not.

We first consider a total energy constraint  $e = 10.8$ . For each iteration step, Figure 3 shows the values of the Lagrange multiplier  $\lambda$ , and the value of the primal objective function QoI (variance).

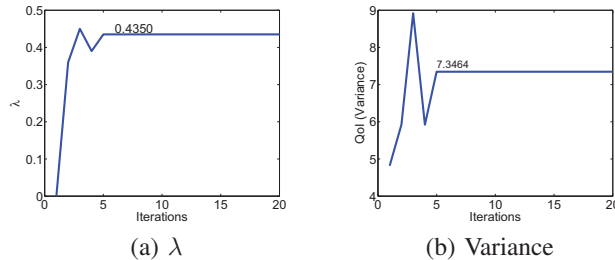


Figure 3. Instance where the primal objective function converges

We initialize  $\lambda = 0$ . We use Algorithm 1 for this experiment, and we choose the step-size at iteration  $k$  to be  $\alpha^{(k)} = 0.05/k$  and  $\gamma$  is set to  $10^{-4}$ . When  $\lambda = 0$ , there are no constraints on the energy, and hence the network uses the maximum possible energy available (which equals 18, as can be easily derived). Since this is greater than the energy constraint  $e = 10.8$ ,  $\lambda$  increases. As  $\lambda$  keeps increasing, the network starts using less energy. Ultimately, it converges to the energy constraint  $e = 10.8$ . Since the QoI is the variance here, when the system uses higher total energy we get a lower variance. These can be seen in Figure 3. We see that both the QoI and  $\lambda$  converge within 5 iterations.

We now consider a total energy constraint  $e = 10$ . We run the same algorithm as before. We observe from Figure 4 that while  $\lambda$  still converges, the primal objective function does not converge. One reason that this happens is because there is no optimal solution that exactly uses the entire permissible energy  $e = 10$  while there are solutions that use a total energy of 7.2 (a permissible solution) and 10.8 (an impermissible solution), respectively. Therefore, we can use the two values of  $\lambda$  corresponding to these energy levels, and determine the optimal solution using other techniques as discussed in Section IV. In this case, it turns out that the optimal solution uses a total energy of 7.2, which corresponds to using all six low energy sensors but no high energy sensor. Note that the behavior of the solution “stabilizes” within 5 iterations.

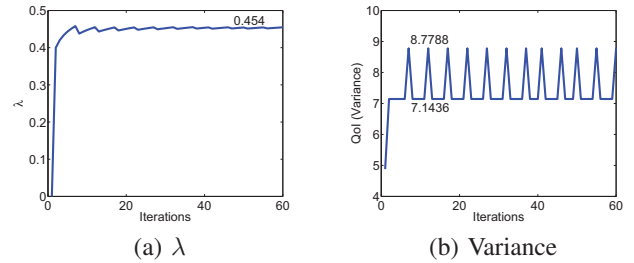


Figure 4. Instance where the primal objective function does not converge

### C. QoI over a Time Horizon

We now study how the QoI behaves over a time horizon for different (average) energy constraint levels and different sampling periods. Note that energy can be saved by a sensor network not only by reducing the total energy used for taking a measurement but also by varying the time period over which a measurement is taken. The targets move according to their mobility models described in the setup. When we do not make a measurement during a time slot (due to the sampling period chosen), we use the predicted variance of the Kalman filter as the QoI. When we make a measurement, we use the corrected variance of the Kalman filter to represent the QoI.

Figure 5 shows the performance of the system over a time horizon for four different values of energy, and three different sampling periods. Note that the energy here is the average energy used over the time horizon. When the sampling period is  $i$  and the average energy used is  $e_{avg}$ , then the energy used during the sampling slot is  $i \cdot e_{avg}$ , and zero during

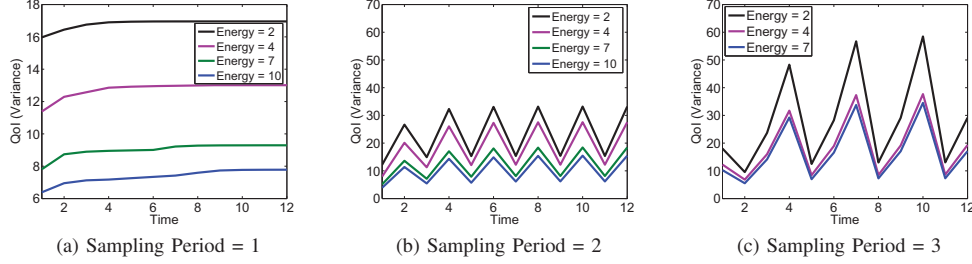


Figure 5. QoI over a time horizon

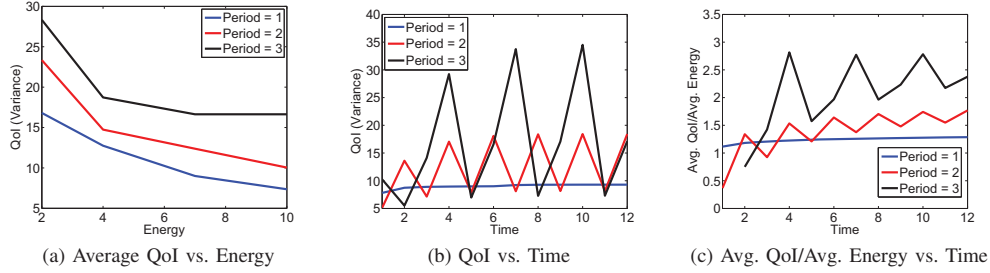


Figure 6. Various QoI performance results

all other slots. From the plots in the figure, we can observe that as the energy increases, the overall variance decreases and hence we get a better QoI. When the sampling period is one, we can see from Figure 5(a) that the overall variance converges in a few time slots. For sampling periods greater than one, we observe that the predicted variance is much higher than the corrected variance. Therefore, when we do not take a measurement we get a high overall variance. This is the reason that we see increased oscillating behavior as the sampling period increases in figures 5(b) and 5(c).

Figure 6(a) represents the QoI averaged over the time horizon for various values of the total energy. It shows that as the sampling period increases, the average QoI becomes worse. This is because the predicted variance is much higher than the corrected variance. It follows from the figure that if there were a choice of reducing energy consumption by either increasing the sampling period or decreasing the energy per slot, we should opt to *lower the energy used per slot while keeping the sampling period low, a combination that improves QoI for the same average energy used.*

Figure 6(b) shows the behavior of the QoI over time for various sampling periods when the average energy over the time horizon is fixed at 7. This figure is particularly interesting because we observe that for the time slots at which we take measurements, the QoI for a higher sampling period actually performs better than the QoI for a lower sampling period. This is because for a higher sampling period, we can use higher energy while still maintaining the average energy at 7. However, because the maximum total energy that can be used by the system is fixed (18 in our case), we cannot arbitrarily keep increasing the sampling period and still get improving QoI. When the sampling period is 3, we use an energy of 21 (which is greater than the maximum total energy available in

the system). Therefore, when the sampling period increases above 3, the variance will again start increasing. For instance, we numerically observed that the variance obtained by the first measurement for sampling period 3 was 5.5151 while that for sampling period 4 was 5.5707. Hence we observe that, *for a given maximum energy  $e_{max}$ , there is a critical sampling period at which the QoI obtained during a sampling slot is minimum. In general, this critical period depends on the system parameters, and in the current case equals to  $\lceil e_{max}/e_{avg} \rceil$ .*

Finally, we compute the ratio of the average QoI attained until the current time slot to the average energy used until the current time slot, and study how it behaves over the time horizon (Figure 6(c)) when the average energy used over the entire time horizon is 7. We observe that the high value of the predicted variance again dictates the behavior of this metric.

#### D. QoI vs. Number of Sensors

In this experiment, we modify the original setup, and study how the QoI varies with the number of sensors, the number of targets, and the level of fusion, i.e., the maximum number of sensors whose measurements were fused. We assume that a sensor can track multiple targets simultaneously. Each sensor uses a bank of Linear Kalman Filters to compute an estimate of the predicted variance for each target (which represents the QoI here). There are no energy constraints in this experiment. The measurement error variance for a sensor is chosen randomly between 0 and 6. The experiment is repeated 300 times, and the results are averaged.

From Figure 7, we see that as the number of sensors increases, the total variance decreases. This is because as the number of sensors increases, on the average, there are more sensors with lower measurement error variances to track targets. However, note that the drop in variance is



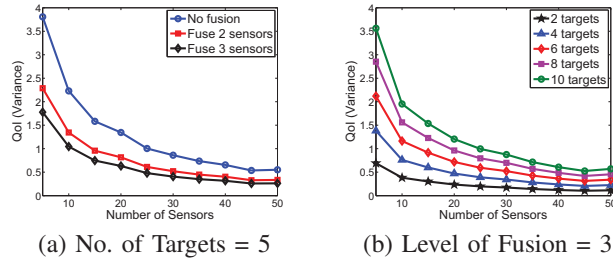


Figure 7. QoI vs. No. of Sensors

significantly lower when the number of sensors increases from, say, 15 to 50 than from 5 to 15. Therefore, *in many applications, it may not be necessary to have a high density of sensors to track a group of targets.*

In Figure 7(a), we fix the number of targets to 5. We vary the levels of fusion, i.e., the maximum number of sensors whose information can be combined. As expected, we get a lower variance with a higher level of fusion. For instance, we get a lower variance when the level of fusion is 3 and there are 15 sensors in the system than when there is no fusion and there are 35 sensors in the system. Therefore, by performing fusion, for the same QoI, we can deploy fewer sensors in the system. Further, we can see that as the number of sensors that are allowed to perform fusion increases, the drop in variance decreases. For instance, there is a higher drop in variance between fusion levels 2 and no fusion than between fusion levels 3 and 2. This means that increasing the fusion level beyond a certain amount in this case does not necessarily result in a significant improvement in performance. In this experiment, we also studied the computational complexity of our algorithm, and observed that it was much lower in practice (using Matlab) compared to our analytical results in Section IV. For instance, for fusion level 3, we observed a complexity of  $O(s^{1.52})$  while our analytical result gives  $O(5s^3)$  (for 5 targets, and no energy constraints). The reason could be that while computing maximum or minimum requires linear worst case complexity, it can be computed in sub-linear time using appropriate data structures.

We now fix the level of fusion to 3, and vary the number of targets in the system (Figure 7(b)). Clearly, as the number of targets increase, the total variance increases. Notice the difference in variance when there are 10 targets in the system, and when there are 2 targets. While this difference is nearly 3 when there are 5 sensors, it is around 0.75 when there are 25 sensors, and 0.5 when there are 50 sensors. Thus, increasing from 25 to 50 sensors only results in a drop of 0.25. This again illustrates the fact that, depending on the system parameters, it may not be necessary to have a high sensor density.

## VI. CONCLUDING REMARKS

We studied the problem of assigning sensors for tracking multiple targets simultaneously. We developed an integer programming methodology for this problem. Compared to existing work, our methodology can assign sensors to track multiple targets, can be used for maximizing a general function of the QoI when information from multiple sensors

are fused, and satisfies energy constraints. In the absence of energy constraints, we showed that this problem can be solved by finding an MWIS, which has a polynomial time complexity because of the structure of the graph in which it is found. We then extended these solutions using a Lagrangian approach when there exist energy constraints.

We provided extensive numerical results applying our methodology to a real multi-target tracking problem, and gained insightful understanding on the convergence of our algorithms, the performance of the QoI over a time horizon, and the effects of varying the sampling period. Finally, we note that while we have considered a specific application of sensor networks (multi-target tracking), our methodology is general enough to be applied to various sensor network operations where fusion plays an important role.

## VII. ACKNOWLEDGMENTS

The authors would like to thank Dr. Thyagaraju Damarla and Dr. Ananthram Swami for insightful discussions.

## REFERENCES

- [1] C. Bisdikian, L. Kaplan, M. Srivastava, D. Thornley, D. Verma, and R. Young, "Building principles for a quality of information specification for sensor information," in *12th International Conference on Information Fusion (FUSION '09)*, 2009.
- [2] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, 2009.
- [3] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "On the use of binary programming for sensor scheduling," *IEEE Transactions on Signal Processing*, 2007.
- [4] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, 2006.
- [5] V. Isler, J. Spletzer, S. Khanna, and T. C. J., "Target tracking with distributed sensors: the focus of attention problem," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
- [6] H. Rowaihy, M. P. Johnson, O. Liu, A. Bar-Noy, T. Brown, and T. La Porta, "Sensor mission assignment in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, 2010.
- [7] A. B. Poore and A. J. R. III, "A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications*, 1997.
- [8] D. J. Papageorgiou and M. R. Salpukas, "The maximum weight independent set problem for data association in multiple hypothesis tracking," *Optimization and Cooperative Control Strategies, Lecture Notes in Computer Science*, 2009.
- [9] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *INFORMS Management Science*, 2004.
- [10] A. Kulik and H. Shachnai, "On lagrangian relaxation and subset selection problems," *Approximation and Online Algorithms, Lecture Notes in Computer Science*, 2009.
- [11] A. Berger, V. Bonifaci, F. Grandoni, and G. Schafer, "Budgeted matching and budgeted matroid intersection via the gasoline puzzle," *Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science*, 2008.
- [12] S. Hariharan, C. Bisdikian, L. M. Kaplan, and T. Pham, "A flexible solutions methodology for resource allocation with applications to sensor network operations," *IBM Res. Rep. RC25115*, Nov. 2010.
- [13] R. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Communications*, 1972.
- [14] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [15] M. Held, P. Wolfe, and H. P. Crowder, "Validation of subgradient optimization," *Mathematical Programming*, 1974.
- [16] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, 1968.